

4 Técnicas de Renderização da Pele Humana

Nesse capítulo são apresentadas e estudadas duas técnicas de renderização da pele humana. Ambas as técnicas utilizam o espaço da textura para simular o espalhamento e absorção da luz através de uma série de *blurs* (suavização através de filtros) em cima da textura de irradiância (textura gerada pela iluminação difusa). A técnica de Gosselin et al. (2004a) é baseada apenas em uma aproximação para representar o efeito do *subsurface scattering*, enquanto que a técnica de d'Eon e Luebke (2007) é baseada em um conjunto de propriedades físicas que descrevem o comportamento da luz na pele humana.

Na técnica de Gosselin et al. (2004a) não é explicitado qualquer algoritmo de reflexão especular, sendo assim, ambas as técnicas estão utilizando o modelo de Kelemen/Szirmay-Kalos (2001) explicado em d'Eon e Luebke (2007). Com isso, a técnica de Gosselin et al. (2004a) apresenta um brilho fisicamente plausível com uma perda de desempenho ínfima.

4.1. Técnica da Convolução de Poisson (Gosselin et al. 2004a)

A idéia de utilizar o espaço da textura e realizar *blurs* para iluminar e reproduzir o *subsurface scattering* da pele surgiu com Borshukov e Lewis (2003) para o filme *The Matrix Reloaded*. O grande trunfo dessa idéia é perceber que o espaço 2D é muito útil e eficiente para simular a natureza local do *subsurface scattering*. A técnica deles baseia-se em renderizar normalmente a pele em uma textura chamada de *Light Map* (ou textura de irradiância) e em cima dessa textura realizar diversos filtros para simular a absorção e iluminação da luz na pele. Gosselin et al. (2004a) utilizou essa idéia e esses conceitos para desenvolver um algoritmo de renderização em tempo real da pele humana na GPU.

Conforme a idéia de Borshukov and Lewis (2003), a técnica de Gosselin et al. (2004a) baseia-se em um *shader* composto de vários passos para realizar a

iluminação difusa no *Light Map*, uma série de *blurs* nessa textura de irradiância e no passo final aplicar a textura resultante dos diversos *blurs* na malha em conjunto com o cálculo de iluminação especular para gerar a imagem final. Antes do início do processo, Gosselin et al. (2004a) utiliza um passo inicial para criar o mapa de sombra do objeto a renderizar (*Shadow Map*).

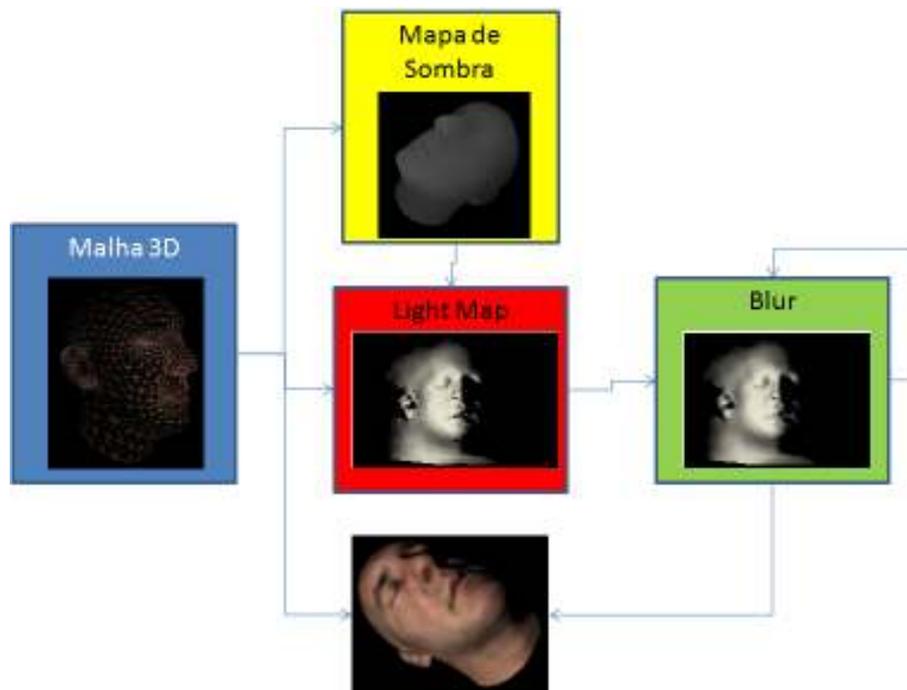


Figura 4 – Algoritmo de iluminação da pele humana de Gosselin et al. (2004a): inicialmente é gerado o mapa de sombra com base na malha 3D, após isso, é criada a textura de irradiância, onde o cálculo difuso de iluminação é feito considerando as sombras. Posteriormente, vários *blurs* são realizados na textura de irradiância, criando assim uma textura final a ser utilizada na malha. Além disso, na última parte, é realizado o cálculo de iluminação especular, para gerar o resultado final.

A técnica de Gosselin et al. (2004a) é resumida pelo seguinte algoritmo (vide Figura 4):

1. Criação do Mapa de Sombra.
2. Renderização da iluminação difusa da pele em uma textura 2D (*Light Map*).
3. *Blurs* no *Light Map* (gera uma textura ao final do processo).
4. Geração da imagem final através da combinação da iluminação especular e da textura gerada ao final do processo de *blur*.

Nas próximas subseções, cada uma das partes desse algoritmo é descrita e detalhada.

4.1.1. Mapa de Sombra

Hoje em dia, existem diversas técnicas para gerar sombra de um objeto em uma cena. Uma das mais simples e antigas que existe é a técnica do mapa de sombra (*Shadow Map* – Williams 1978 citado por Möller et al. 2008). A ideia por trás dessa técnica é bastante simples: inicialmente renderiza-se uma cena do ponto de vista da luz e armazena-se em uma textura (mapa de sombra) a distância para a luz de cada ponto iluminado. Posteriormente, quando a cena é renderizada tendo outra posição qualquer do espaço como ponto de vista, para cada ponto da malha visualizado é calculada a distância do mesmo em relação à luz. Se a distância for maior do que a distância armazenada no mapa de sombra, o ponto não é iluminado (por aquela fonte de luz), caso contrário, o ponto é iluminado normalmente. A Figura 5 exemplifica a técnica do *Shadow Map*, enquanto que a Figura 6 mostra um exemplo de mapa de sombra gerado.

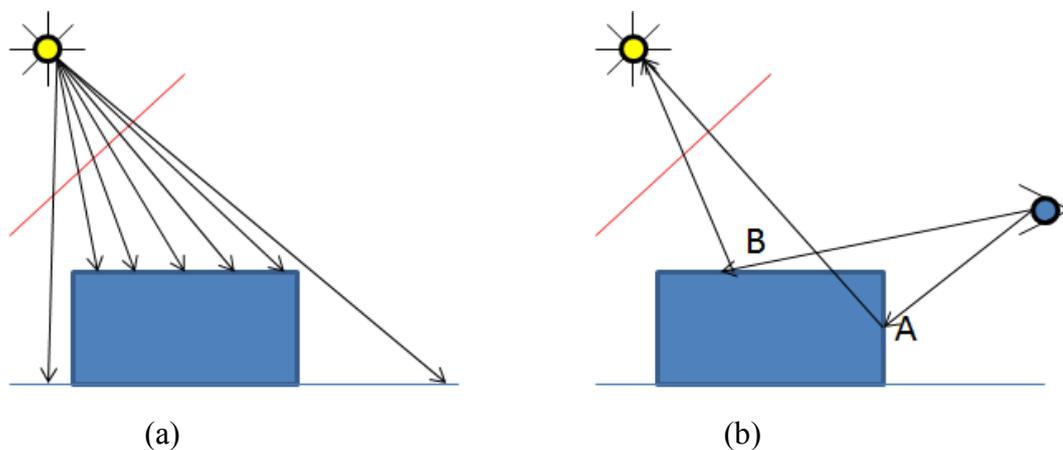


Figura 5 – A imagem à esquerda (a) representa a cena sendo renderizada tendo como ponto de vista a fonte de luz. No mapa de sombra (linha vermelha) é armazenada a distância de cada ponto do objeto sendo iluminado. Na imagem à direita (b), a cena é renderizada tendo o observador como ponto de vista. O ponto B visto pelo observador é iluminado, pois a distância do ponto à fonte de luz é igual (ou muito próxima) da distância armazenada no mapa de sombra. Já o ponto A está em sombra, pois a distância desse ponto à fonte de luz é bem maior do que a distância armazenada no mapa de sombra.

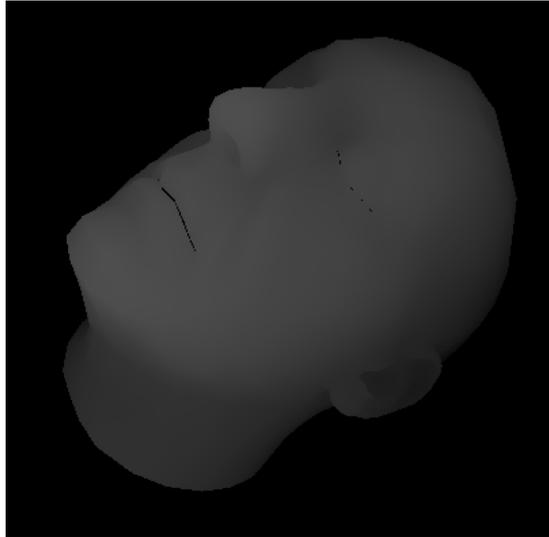


Figura 6 – Exemplo de mapa de sombra de uma malha iluminada por um ponto de luz. Nesse mapa é armazenada a distância de cada ponto para a luz.

Para gerar o mapa de sombra o seguinte algoritmo deve ser realizado a cada *frame*:

1. Gera-se a matriz de visualização *View*, conforme a equação (1), a qual coloca a luz como ponto de vista para a renderização.
2. Gera-se a matriz de projeção *P*, conforme a equação (5), de forma que o espaço de projeção ocupe praticamente o mesmo espaço da caixa envolvente do objeto renderizado, gerando assim resultados mais precisos.
3. Gera-se a posição do ponto p' (vide equação (6)) no espaço projetivo tendo a luz como ponto de vista. Para isso, basta multiplicar a posição do ponto p do objeto pela matriz do espaço do mundo M , pela matriz de visualização *View* e pela matriz de projeção *P*.
4. Por fim, armazena-se na textura final o resultado da divisão da coordenada z pela coordenada w de p' .

$$View = \begin{pmatrix} X \cdot x & Y \cdot x & Z \cdot x & 0 \\ X \cdot y & Y \cdot y & Z \cdot y & 0 \\ X \cdot z & Y \cdot z & Z \cdot z & 0 \\ - (X, L) & - (Y, L) & - (Z, L) & 1 \end{pmatrix} \quad (1)$$

onde X, Y, Z são os eixos de coordenadas da luz e L a posição da luz no espaço do mundo. X, Y e Z são calculados pelas equações (3), (4) e (2), respectivamente (UP é o vetor para cima do ponto de vista da luz).

$$Z = - \frac{L}{\|L\|} \quad (2)$$

$$X = - \frac{Up \times Z}{\|UP \times Z\|} \quad (3)$$

$$Y = Z \times X \quad (4)$$

$$P = \begin{pmatrix} \frac{h}{\operatorname{tg}\left(\frac{fov}{2}\right) * w} & 0 & 0 & 0 \\ 0 & \frac{1}{\operatorname{tg}\left(\frac{fov}{2}\right)} & 0 & 0 \\ 0 & 0 & \frac{far}{far - near} & 1 \\ 0 & 0 & \frac{-far * near}{far - near} & 0 \end{pmatrix} \quad (5)$$

onde w e h são as dimensões da região de projeção, fov é o ângulo de abertura, $near$ e far são as distâncias dos planos para o ponto de luz.

$$p' = (((p * M) * View) * P) \quad (6)$$

Pela simplicidade, essa técnica possui alguns pontos falhos. Se a precisão do z -buffer não for boa, assim como a resolução do mapa em pixels não for grande o suficiente para a malha analisada, falsos positivos são gerados na hora de comparar a distância de um ponto com a distância armazenada no mapa de sombra. Além disso, sombras “fortes” e com *aliasing* são geradas, pois não há nenhuma suavização ou meio termo entre um ponto iluminado e não iluminado (Möller et al. 2008). Gosselin et al. (2004a) e Möller et al. (2008), mostram que o primeiro problema pode ser amenizado (ou até mesmo resolvido) utilizando um valor de tolerância (tol) constante no momento de comparar as distâncias, conforme exemplifica a equação (7).

$$\begin{cases} \text{sombra se } d(P) - ds > tol \\ \text{iluminado se } d(P) - ds \leq tol \end{cases} \quad (7)$$

onde $d(P)$ é a distância do ponto verificado até a fonte de luz e ds é a distância armazenada no mapa de sombra.

O segundo problema é resolvido sem nenhum custo adicional com o uso da técnica de Gosselin et al. (2004a). Com o *blur* realizado para simular o *subsurface scattering* da pele, as sombras são automaticamente suavizadas, gerando assim resultados mais realistas, conforme mostra a Figura 7.

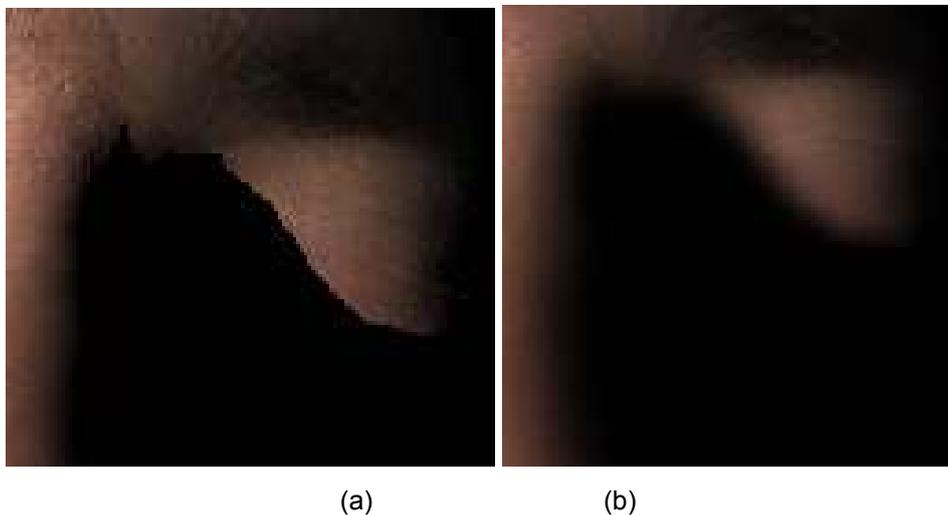


Figura 7 – A imagem à esquerda (a) mostra o resultado da aplicação do uso da técnica do *Shadow Map*. Pode ser visto que uma sombra forte é criada. A imagem à direita mostra o resultado da sombra gerada após os *blurs* da textura de irradiância. Nota-se que a sombra fica mais suave e o resultado mais parecido com a realidade.

A técnica do *Shadow Map* pode ser utilizada para várias fontes de luz. Para isso, deve ser feito um mapa de sombra para cada uma dessas fontes. A comparação se um ponto é iluminado ou não deve ser feita levando em consideração cada um desses mapas. O processo de comparação (o qual cria efetivamente as sombras nos objetos) é realizado durante o passo de geração do mapa de irradiância, conforme descrito na próxima seção.

4.1.2. Iluminação Difusa (Mapa de Irradiância ou *Light Map*)

A segunda etapa do algoritmo de Gosselin et al. (2004a) é a realização da iluminação difusa da pele em uma textura 2D conhecida como mapa de irradiância ou *Light Map*.

Conforme mencionado anteriormente, esse algoritmo utiliza o espaço da textura para realizar a iluminação e *subsurface scattering* da pele, logo é necessário fazer o *unwrap* da geometria do espaço tridimensional para o espaço bidimensional da textura. Green (2004) utilizou a idéia de Borshukov e Lewis (2003) e demonstrou como realizar esse procedimento na GPU de forma bastante simples: ao usar as coordenadas de textura como coordenadas do objeto para fazer a projeção do *vertex shader* para o *fragment shader* (*pixel shader*), mapeia-se toda a superfície da geometria independente da posição da câmera. Como as coordenadas de textura têm pontos no intervalo [0,1], deve ser feita a conversão para o espaço [-1,1] (espaço projetivo). Para isso basta realizar o cálculo exibido na equação (8).

$$pos.xy = coord_Tex * 2 - 1 \quad (8)$$

Após realizar o *unwrap* da malha para o espaço da textura, é feito o cálculo da iluminação difusa propriamente dita. Tal cálculo é feito de acordo com a equação (9).

$$difuse = L_c * (N.L) \quad (9)$$

onde L_c é cor da luz, N é o vetor normal e L é o vetor da direção da fonte de luz ao ponto da geometria.

Na seção anterior é mostrado como o mapa de sombra é gerado e que essa textura é usada para criar as sombras na pele. Pela equação (7) pode-se calcular se um ponto está em sombra ou iluminado, bastando apenas determinar $d(P)$ para realizar o cálculo de comparação. Tal variável é calculada seguindo o mesmo algoritmo de geração do *Shadow Map* sem a parte de armazenar o resultado em

uma textura. Dessa forma, a iluminação difusa final é feita com base na equação (10).

$$final_difuse = fac_shadow * difuse \quad (10)$$

onde o fator de sombra fac_shadow é 0 se o ponto está em sombra ou 1 se está iluminado.

Ao final do processo, é gerada uma textura com a iluminação difusa da pele, com sombras, da geometria analisada. A Figura 8 mostra um exemplo de mapa de irradiância gerado.



Figura 8 – Exemplo de mapa de irradiância com sombras.

4.1.3. O Processo de Convolução de Poisson (*Blurs*)

Matematicamente falando, convolução é o processo de realizar uma operação entre duas funções e obter uma terceira a qual é considerada uma versão modificada das versões originais. No contexto dessa dissertação, convolução consiste na aplicação de um filtro (primeira função – representa uma distribuição probabilística, estatística, entre outras) em uma imagem (segunda função) para

gerar outra imagem (versão modificada da versão original) de forma a se obter um determinado resultado desejado.

A etapa de convolução da textura de irradiância pelos filtros de Poisson (modelo estatístico de distribuição de eventos) é a parte mais importante do algoritmo de Gosselin et al. (2004a). Nessa etapa é realizada a simulação do comportamento de espalhamento e absorção da luz pela pele humana.

Nessa parte do processo são realizados vários *blurs* da textura de irradiância com base em um filtro bidimensional de Poisson com 12 pontos de amostragem. Tal procedimento é realizado várias vezes pelo mesmo *pixel shader* tendo como entrada primeiramente a textura de irradiância e nas demais vezes a textura gerada pela convolução resultante do passo anterior.

A realização de cada *blur* é feita pelo seguinte procedimento:

1. Obtém-se o pixel da textura que se deseja simular o *subsurface scattering*.
2. Calcula-se o tamanho de cada pixel da textura tanto para largura quanto para o comprimento (conforme a equação (11)). Esse cálculo é feito para determinar o tamanho do pixel no espaço da textura [0,1].
3. Calcula-se o raio de atuação da distribuição de Poisson em volta do pixel analisado (conforme a equação (12)).
4. Calcula-se o valor do pixel na nova textura com base nos 12 pontos vizinhos obtidos através da distribuição de Poisson (conforme equação a (13)).

$$size = 1/(width \text{ ou } height) \quad (11)$$

$$radius = size * (kernel_size * scale_x + scale_y) \quad (12)$$

onde *kernel_size* é um valor auxiliar gerado por um artista (vide seção 5.3) e *scale* é um valor de escala auxiliar utilizado para aumentar ou diminuir o tamanho do raio de atuação.

$$pixel = \frac{pixel_color(coord)}{13} + \frac{\sum_{i=0}^{11} pixel_color(coord + poisson(i)*radius)}{13} \quad (13)$$

onde $pixel_color(x)$ é a cor do pixel na coordenada x da textura analisada e $poisson(i)$ é um dos 12 valores de pontos de Poisson utilizados para a convolução.

O uso do $kernel_size$ por um artista permite variar o tamanho do raio do filtro de Poisson dependendo da parte analisada da pele. Regiões como a orelha e o nariz têm maior espalhamento, enquanto que regiões como as bochechas sofrem menos esse efeito (Gosselin 2004, Gosselin et al. 2004b, Oat 2004).

Ao final do procedimento, é criada uma textura “embaçada” que simula os efeitos do *subsurface scattering* na pele humana (vide Figura 9). Essa textura é utilizada no último passo do algoritmo para gerar a imagem final exibida ao usuário. Além disso, conforme descrito na seção 4.1.1, esse procedimento suaviza e deixa mais realista as sombras visualizadas na pele humana.



Figura 9 – Exemplo de textura gerada após algumas convoluções de Poisson realizadas na textura de irradiância.

4.1.4. Reflexão Especular

Antes de entrar na parte final do algoritmo de Gosselin et al. (2004a) é importante explicar a técnica de reflexão especular utilizada nessa dissertação. Gosselin et al. (2004a) não fixa técnica de reflexão especular em seu trabalho, assim sendo, é utilizado o modelo de Kelemen/Szirmay-Kalos (2001) explicado em d'Eon e Luebke (2007) para representar o brilho visto na pele humana. Essa técnica tem como vantagens a fácil implementação e bom desempenho, além de ser baseada em um modelo fisicamente plausível.

A equação (16) mostra como o modelo de Kelemen/Szirmay-Kalos (2001) é formulado. Como todo modelo de reflexão especular fisicamente plausível, este contém um termo de Fresnel F , representado pela aproximação de Schlick (1993 citado por d'Eon e Luebke 2007) na equação (14). Além disso, a equação (16) é composta por um termo B o qual representa a função de distribuição de Beckmann. A função de Beckmann é um dos modelos de funções probabilísticas que representa a distribuição das normais das pequenas microgeometrias que formam as superfícies dos objetos. Essas pequenas microgeometrias são conhecidas como microfacetas e representam pequenas camadas finas que agem como um espelho refletor (Torrance e Sparrow 1967 citado por Möller et al. 2008). O termo B é calculado de acordo com a equação (15).

$$F = (1 - \cos(V, H))^5 + F_0(1 - (1 - \cos(V, H))^5) \quad (14)$$

$$B = \frac{1}{m^2 \cos^4(N, H)} e^{\frac{-\text{tg}^2(\arccos(N, H))}{m^2}} \quad (15)$$

$$\text{specular} = \frac{BF}{\cos^2(L+V, L+V)} \quad (16)$$

onde V é o vetor de direção do observador, H é o vetor bissetriz entre L e V , m é a rugosidade do material e F_0 é o parâmetro da função de reflexão de Fresnel que representa um valor de reflexão quando o raio de luz é perpendicular à superfície. Esse valor é característico de cada material (0.028 para a pele assumindo que o

índice de refração da mesma é de 1.4 (Donner e Jensen 2006 citado por d'Eon 2007b)).

O denominador da equação (16) representa a possibilidade que um raio de luz tem de ser refletido na direção do observador sem estar em sombra ou mascarado. Nesse ponto está a vantagem de utilizar o modelo de Kelemen/Szirmay-Kalos (2001) ao invés do modelo de Cook e Torrance (1981 citado por Möller et al. 2008). Esse termo é mais estável e mais eficientemente calculado.

A Figura 10 mostra o resultado do uso dessa técnica para gerar o brilho na pele humana.

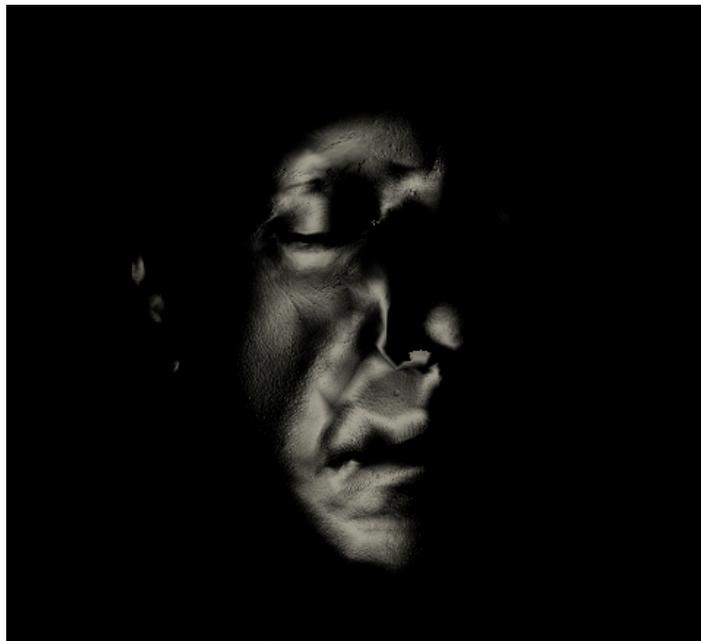


Figura 10 – Exemplo de reflexão especular com o uso da técnica de Kelemen/Szirmay-Kalos (2001). Nessa imagem o contraste foi alterado para facilitar a visualização.

4.1.5. Renderização Final da Malha

Na parte final do algoritmo de Gosselin et al. (2004a) é feita a junção das reflexões especular e difusa mostradas anteriormente para gerar a imagem final da pele humana. Nessa etapa, portanto, volta-se a trabalhar no espaço tridimensional, onde a textura obtida dos *blurs* do *Light Map* é aplicada à malha e o cálculo de reflexão especular é realizado de forma a obter o resultado final.

Até essa etapa, a cor da pele é desconsiderada de qualquer cálculo preliminar. Todas as convoluções realizadas consideram apenas a interação e cor da luz com o modelo. Para gerar a imagem final, deve-se então agregar a cor da pele no ponto observado com o espalhamento da luz ocorrido nessa região. Assim sendo, o termo difuso final é dado pela equação (17).

$$final_difuse_color = C * final_difuse_blurred \quad (17)$$

onde C é a cor da pele e $final_difuse_blurred$ é a cor difusa após as convoluções realizadas no *Light Map*.

As sombras calculadas com base no *Shadow Map* são utilizadas apenas para gerar o mapa de irradiância e assim compor o termo difuso da pele. Porém, se não forem consideradas as sombras na hora de calcular o termo especular final, a imagem resultante pode apresentar brilhos em regiões nas quais a luz não chega. Dessa forma, Gosselin et al. (2004a) propõem utilizar a luminância do mapa de irradiância após as convoluções para atenuar o termo especular nessas regiões, criando assim uma aparência mais natural ao brilho da pele. Assim sendo, utilizando o termo especular de Kelemen/Szirmay-Kalos (2001) com a idéia de atenuação do brilho de Gosselin et al. (2004a), a componente final especular é calculada pela equação (18).

$$final_specular_color = specular * L_c * (LUM. \\ final_difuse_blurred) \quad (18)$$

onde LUM são os termos de luminância para uma cor no RGB (0.2125, 0.7154, 0.0721).

Assim sendo, a cor final de cada pixel é dada pela equação (19).

$$final_color = final_specular_color + final_difuse_color \quad (19)$$

A Figura 11 mostra um exemplo de resultado obtido ao final de todo o algoritmo de Gosselin et al. (2004a).

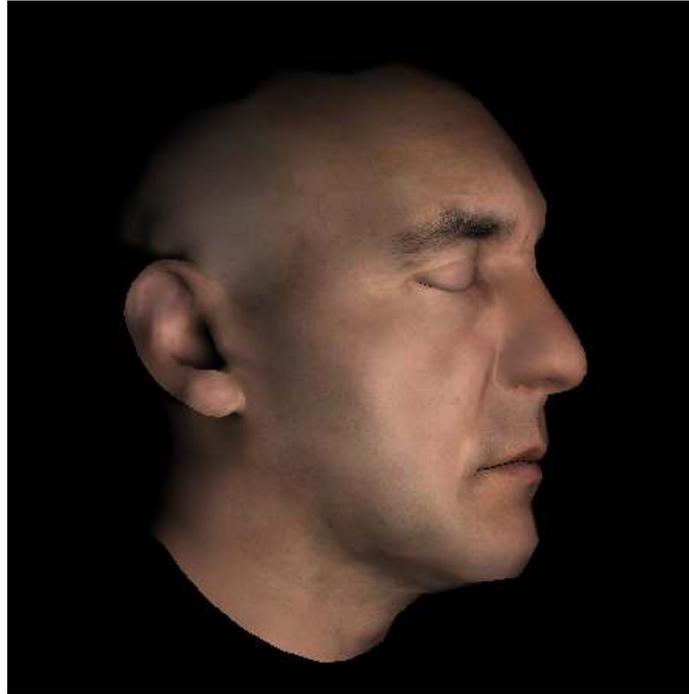


Figura 11 – Exemplo de pele humana gerada com o algoritmo de Gosselin et al. (2004a). A imagem apresenta o resultado da junção das componentes difusa e especular.

Nos capítulos 5 e 6 são apresentados mais resultados do uso dessa técnica, assim como vários detalhes de implementação e um conjunto de vantagens e desvantagens observadas.

4.2. Técnica dos Perfis de Difusão (d'Eon e Luebke 2007)

O algoritmo de d'Eon e Luebke (2007) de renderização de pele humana em tempo real é considerado o estado da arte das técnicas existentes atualmente com esse objetivo. O principal motivo disso deve-se ao conjunto de conceitos físicos sobre o comportamento da luz na pele que são utilizados para formar a técnica.

Apesar dos conceitos por trás do algoritmo serem mais fiéis com a realidade, a idéia geral da técnica em si parece muito com a técnica de Borshukov e Lewis (2003) e Gosselin et al. (2004a): é feita a iluminação difusa da malha em uma textura (textura de irradiância), uma série de convoluções (*blurs*) nessa textura (para simular o *subsurface scattering*) e ao final é realizada a aplicação das texturas geradas sobre o objeto somadas com a reflexão especular. A Figura 12 mostra um esquemático do algoritmo de d'Eon e Luebke (2007).

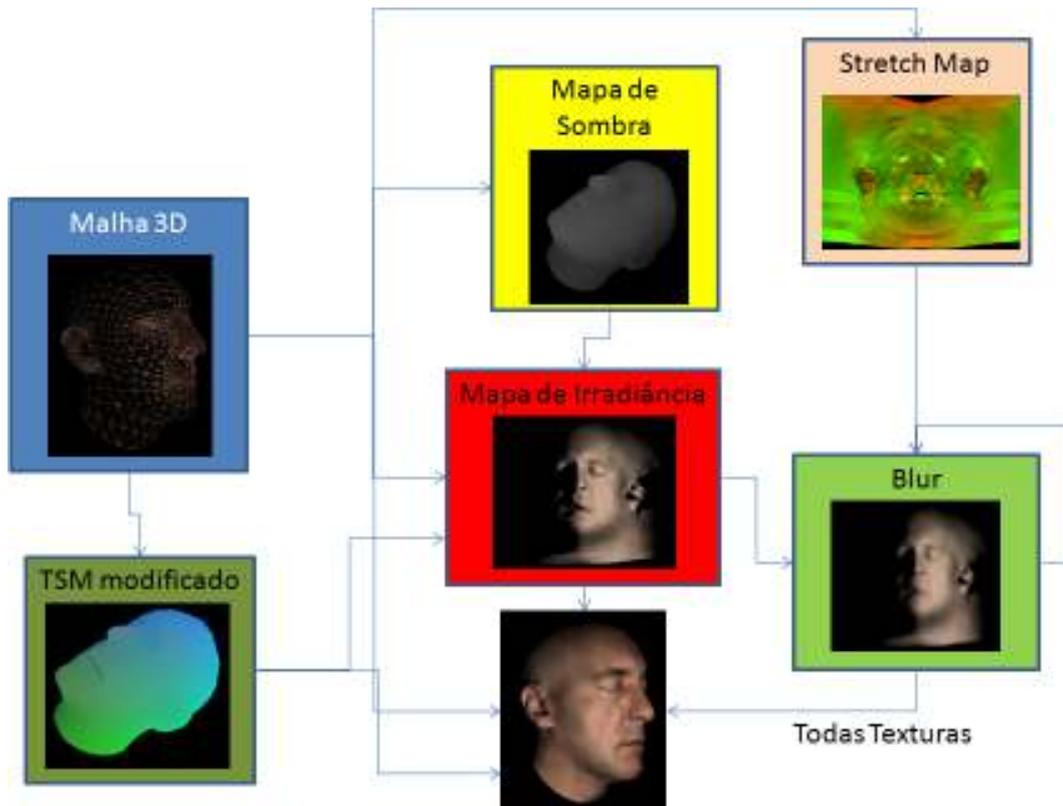


Figura 12 – Visão geral do algoritmo de d'Eon e Luebke (2007) para renderização de pele humana.

A técnica de d'Eon e Luebke (2007) pode ser resumida pelo seguinte procedimento:

1. Cria-se o Mapa de Sombra.
2. Cria-se o mapa de sombra translúcida modificado (*TSM modificado*), o qual é utilizado para representar o espalhamento global da luz na pele.
3. Cria-se o *Stretch Map* (para correção das distorções provocadas nas distâncias dos pontos do espaço 3D para o 2D devido ao *unwrap* da malha).
4. Renderiza-se a iluminação difusa da pele em uma textura 2D (mapa de irradiância).
5. Realizam-se várias convoluções do mapa de irradiância de acordo com o perfil de difusão da luz na pele (cada convolução gera uma nova textura).
6. Gera-se a imagem final através da combinação linear das texturas do passo anterior mais o cálculo de iluminação especular.

Nas próximas seções cada passo da técnica é detalhado, com exceções da criação do *Shadow Map*, que funciona de forma semelhante à explicação da seção 4.1.1 (a técnica de d'Eon e Luebke (2007) também suaviza as sombras pelos processos de convolução), e do algoritmo de reflexão especular utilizado, que é o mesmo explicado na seção 4.1.4. Porém, antes de entrar na explicação de cada passo, é feito um detalhamento da teoria dos perfis de difusão, na qual é baseada toda a parte de espalhamento e absorção da luz na pele da técnica de d'Eon e Luebke (2007).

4.2.1. Perfis de Difusão

Os perfis de difusão representam uma aproximação da maneira na qual a luz se espalha abaixo da superfície de determinados materiais translúcidos e emerge novamente. Ao emitir um raio de luz de forma perpendicular a uma mesa em uma sala escura, pode-se verificar que a região ao redor do ponto de entrada da luz na superfície também fica iluminada. Essa iluminação vai se enfraquecendo à medida que se afaste ao ponto de entrada. O raio de iluminação ao redor do ponto de entrada é diferente para cada material analisado, sendo, assim, uma característica do mesmo. A função que determina o raio de espalhamento da luz na superfície analisada é chamada de perfil de difusão do material. O ângulo em que o raio atinge a superfície influi na direção do espalhamento, porém se for considerado que o material é homogêneo, o espalhamento é igual em todas as direções, logo o ângulo torna-se um fator irrelevante. A Figura 13 mostra um exemplo de feixe de luz entrando em uma superfície e se espalhando ao redor do ponto de entrada.

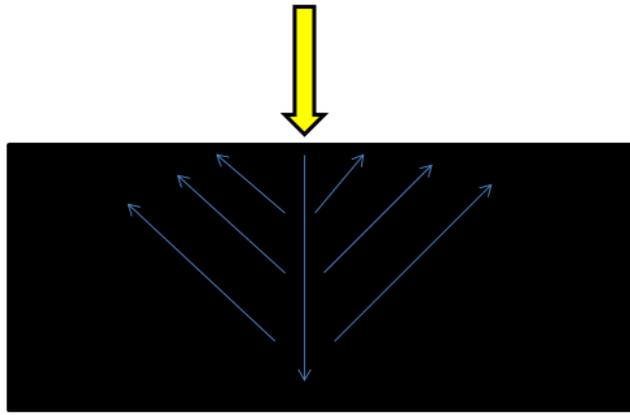


Figura 13 – Um feixe de luz ao entrar em uma superfície espalha-se ao redor do ponto de entrada, iluminando as regiões (radialmente) ao redor desse ponto.

Os perfis de difusão são diferentes para cada cor analisada, tendo cada uma sua própria função característica. O espectro vermelho se espalha mais que os espectros verde e azul nos materiais. Isso explica, porque um material iluminado pode apresentar nas partes mais afastadas do foco de luz bordas mais avermelhadas. A Figura 14 mostra um exemplo de como as cores azul, verde e vermelha (as cores usadas no RGB, sendo assim foco desse trabalho) tendem a se comportar e espalhar abaixo da superfície de determinados materiais.

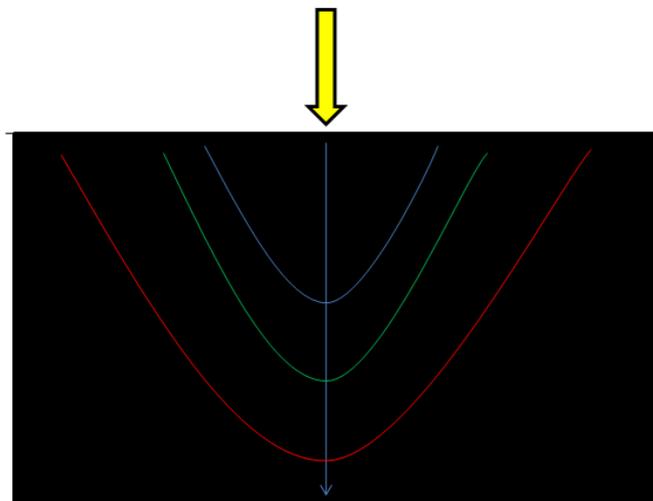


Figura 14 – Exemplo das regiões afetadas pelo espalhamento das cores azul, verde e vermelha ao redor de um ponto em um determinado material.

Com base nesses conceitos, pode-se imaginar cada pequena região da pele como se fosse o experimento do feixe de luz na mesa descrito anteriormente (d'Eon e Luebke 2007). Assim, ao invés de se calcular o espalhamento de cada

feixe luminoso sob as camadas internas da pele humana, pode-se usar um perfil de difusão que aproxime o espalhamento da luz na pele humana.

Os perfis de difusão ($R(r)$) que representam determinados materiais geralmente são funções analíticas bastante complexas e de difícil reprodução. No entanto, essas funções podem ser aproximadas através da soma de várias gaussianas ($G(v,r)$) como exemplificado nas equações (20) e (21) (d'Eon e Luebke 2007). Encontrar o número de gaussianas que representa de forma satisfatória o perfil de difusão de um determinado material não é uma tarefa simples e baseia-se em um processo de minimização de erros, conforme a equação (22).

$$G(v,r) = \frac{1}{2\pi v} e^{-\frac{r^2}{2v}} \quad (20)$$

onde v é a variância e r é a distância radial ao foco de luz.

$$R(r) \approx \sum_{i=1}^k w_i G(v_i, r) \quad (21)$$

onde w é o peso de cada gaussiana e k é o número de gaussianas usadas para representar o perfil de difusão.

$$\int_0^{\infty} r \left(R(r) - \sum_{i=1}^k w_i G(v_i, r) \right)^2 dr \quad (22)$$

Conforme mencionado no capítulo 3, considerar três camadas de pele para modelar o comportamento da luz ao entrar no corpo, é suficiente para produzir resultados satisfatórios (Donner e Jensen 2005 citado por d'Eon e Luebke 2007). d'Eon e Luebke (2007) mencionam que o uso de seis gaussianas forma uma boa aproximação para o perfil de difusão que representa o espalhamento e absorção de luz na pele para esse modelo de três camadas. A Tabela 2 mostra os parâmetros das gaussianas para obter a aproximação desse perfil (para um homem caucasiano) para cada uma das componentes RGB.

Tabela 2 – Parâmetros das seis gaussianas que formam o perfil de difusão para as cores vermelha, verde e azul de um homem caucasiano (d'Eon e Luebke 2007).

Peso/Variância das Gaussianas para cada Comprimento de Onda			
Variância (mm ²)	Vermelho	Verde	Azul
0.0064	0.233	0.455	0.649
0.0484	0.1	0.336	0.344
0.187	0.118	0.198	0
0.567	0.113	0.007	0.007
1.99	0.358	0.004	0
7.41	0.078	0	0

Com base na teoria apresentada, pode-se explicar os passos do algoritmo de d'Eon e Luebke (2007), principalmente a etapa dos *blurs* que representa o perfil de difusão da luz na pele.

4.2.2. Iluminação Difusa (Mapa de Irradiância)

A iluminação difusa do algoritmo de d'Eon e Luebke (2007) funciona de forma semelhante ao algoritmo de Gosselin et al. (2004a) descrito na seção 4.1.2. Porém, d'Eon e Luebke (2007) apresentam algumas poucas diferenças que servem para aumentar ainda mais o realismo da técnica como a conservação da energia luminosa que chega a um ponto assim como a transmissão da luz na pele em regiões finas como orelhas e nariz. As melhorias na parte de iluminação difusa e geração do mapa de irradiância são explicadas nas seções 4.2.5 à 4.2.7.

4.2.3. Convoluções da Textura de Irradiância

Assim como no algoritmo de Gosselin et al. (2004a), essa é a etapa mais importante do algoritmo de d'Eon e Luebke (2007), pois é nessa parte que são geradas as peças-chaves para simular o comportamento de espalhamento e absorção da luz pela pele humana.

Para se reproduzir tal comportamento, é necessário simular o perfil de difusão que representa o *subsurface scattering* da luz na pele. Essa simulação

começa através da convolução da textura de irradiância por cada uma das gaussianas que formam esse perfil. Cada convolução gera uma textura resultante, a qual é multiplicada pelo seu peso correspondente na Tabela 2 e, ao final, as resultantes dessas multiplicações são somadas reproduzindo assim o *subsurface scattering*. Nessa etapa do algoritmo são geradas apenas as texturas por gaussiana, ficando a parte da combinação linear dessas para a última etapa do algoritmo (seção 4.2.8).

Conforme pode ser visto na Tabela 2, existem seis gaussianas para cada componente R, G e B para simular o espalhamento da luz na pele humana. Sendo assim, são necessárias seis convoluções por componente. Segundo d'Eon e Luebke (2007), como o espaço afetado pela primeira gaussiana é muito curto (vide a variância de 0.0064 mm^2), o *blur* da textura da irradiância praticamente não altera o resultado, logo, a própria textura de irradiância pode ser utilizada como primeira convolução. Isso diminui o número de *blurs* a serem realizados (cinco por componente). Como o cálculo das três componentes pode ser realizado ao mesmo tempo na GPU por gaussiana, ao final do processo são geradas cinco texturas além da textura de irradiância.

A geração da imagem final de difusão da luz na pele pode ser feita pela combinação linear das gaussianas devido às propriedades específicas que as convoluções têm. De fato, o correto é convoluir a imagem pelo perfil de difusão inteiro e não por cada uma das gaussianas. Porém a convolução de uma imagem pela soma de funções é igual à soma da convolução da imagem por cada uma das funções. Assim o *blur* do mapa de irradiância pelo perfil de difusão da luz na pele pode ser escrito de acordo com a equação (23).

$$I * R(r) \approx I * \left(\sum_{i=1}^k w_i G(v_i, r) \right) = \sum_{i=1}^k w_i I * G(v_i, r) \quad (23)$$

onde I é a imagem a ser convoluída.

Convoluir a imagem pelo perfil de difusão $R(r)$ consiste em um processo de aplicar um filtro bidimensional em cada ponto onde $N \times N$ acessos à textura são realizados. Esse processo é custoso dependendo do tamanho desse filtro. d'Eon e Luebke (2007) mostram que N pode chegar até 64 para representar todo o

espalhamento da componente vermelha, gerando assim 4096 acessos à textura por pixel.

As gaussianas têm a propriedade de serem separáveis. Logo, se o perfil de difusão pudesse ser representado por apenas uma gaussiana, o filtro de convolução poderia ser modelado de forma unidimensional e ser aplicado uma vez na direção horizontal e outra na direção vertical para obter o mesmo resultado. Esse processo geraria apenas $N + N$ acessos à textura.

Essa propriedade das gaussianas é importante para diminuir a quantidade de acessos à textura realizados no cálculo dos *blurs* no *fragment shader*. Um filtro 10 x 10 qualquer, por exemplo, normalmente faz 100 acessos à textura para gerar o resultado esperado por esse processamento em um único passo. Porém, se o filtro for gaussiano, pode-se realizar, em dois passos, 10 acessos à textura na direção vertical mais 10 acessos à textura na direção horizontal para gerar o mesmo resultado. Nesse exemplo, o número de acessos à textura diminui em 80 com o custo de apenas o uso de um passo a mais.

Como visto anteriormente, o perfil de difusão não pode ser representado por apenas uma gaussiana, mas pode por uma combinação dessas. Dessa forma, cada convolução do mapa de irradiância pelas gaussianas da Tabela 2 pode ser feita em duas partes no algoritmo: uma na direção vertical e outra na direção horizontal. A propriedade da separação é o principal motivo pelo qual d'Eon e Luebke (2007) utilizam gaussianas para aproximar o perfil de difusão da luz na pele.

d'Eon e Luebke (2007) sugerem sete pontos para representar o filtro unidimensional a ser aplicado a cada direção de convolução. Esse filtro é usado para todos os *blurs*, devendo apenas ser escalado linearmente pelo desvio padrão da gaussiana a ser aplicada. Todo o processo de simulação do *subsurface scattering* realiza assim $6 * 2 * 7 = 84$ acessos à textura (6 gaussianas, 2 direções e 7 pontos representativos) por pixel, ao invés de $64 * 64 = 4096$ acessos.

Outra propriedade que d'Eon e Luebke (2007) usufruem das gaussianas é o fato que a convolução de duas gaussianas gera outra gaussiana. Essa propriedade também é importante, pois se pode gerar cada textura resultante através da convolução do resultado de uma anterior. Isso permite, assim, que a textura original seja convoluída por gaussianas maiores sem a necessidade de aumentar o tamanho do filtro de sete pontos representativos a cada passo do algoritmo. Portanto, se o resultado anterior de uma convolução de uma textura I por uma

gaussiana com variância v_1 for I' e for desejado obter o resultado da convolução de I por uma gaussiana v_2 , basta fazer a convolução de I' por v_2-v_1 . Essa propriedade é formulada pela equação (24).

$$G(v_1, r) * G(v_2, r) = G(v_1 + v_2, r) \quad (24)$$

Finalizando, cada etapa do processo de *blur* de d'Eon e Luebke (2007) pode ser descrito pelo seguinte passo a passo:

1. Obtém-se o pixel da textura que se deseja simular o *subsurface scattering*.
2. Calcula-se o tamanho de cada pixel da textura tanto para largura quanto para o comprimento (conforme equação (11)). Esse cálculo é feito para determinar o tamanho do pixel no espaço da textura [0,1].
3. Calcula-se o raio de atuação da gaussiana em volta do pixel analisado (conforme equação (25)). A equação (25) possui um termo *stretch* o qual corrige o raio da gaussiana de acordo com a distorção provocada pelo *unwrap* da malha. A seção 4.2.4 explica mais detalhadamente essa correção.
4. Calcula-se o valor do pixel na nova textura com base nos 7 pontos da gaussiana (conforme equação (26)).

$$radius = size * stretch_x * \alpha \quad (25)$$

onde *stretch.x* é a correção de distorção do *unwrap* da malha na direção horizontal e α é o desvio padrão da gaussiana. Quando o processo é realizado na direção vertical, utiliza-se *stretch.y*.

$$pixel = \sum_{i=-3}^3 pixel_color(coord.x + i * radius) * gaussian(i) \quad (26)$$

onde *gaussian(i)* é um dos 7 valores de pontos do filtro unidimensional utilizado para a convolução. Para a direção horizontal utiliza-se *coord.x* e para a direção vertical utiliza-se *coord.y*.

O processo descrito acima deve ser feito para cada gaussiana e para as direções vertical e horizontal. As Figuras 15 e 16 mostram exemplos de resultados obtidos com a convolução da textura de irradiância por gaussianas diferentes.



Figura 15 – Exemplo de resultado obtido pela primeira convolução (representando a segunda gaussiana) da textura de irradiância.

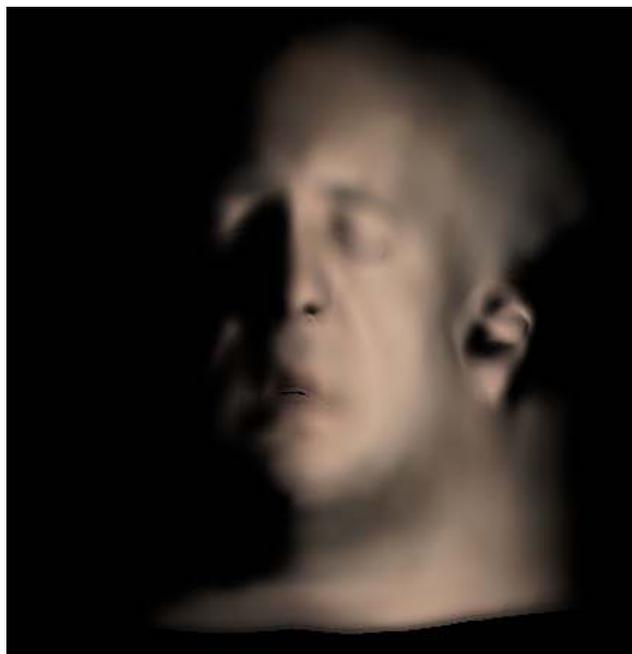


Figura 16 - Exemplo de resultado obtido após todas as convoluções (representando a sexta gaussiana) da textura de irradiância.

4.2.4. Correção da Distorção do *Unwrap*

Embora facilite simular o espalhamento da luz através das convoluções das texturas, o processo de *unwrap* de uma malha não mantém necessariamente uma correspondência direta entre as distâncias de dois pontos na geometria e a distância dos mesmos no espaço da textura. Esse problema ocorre devido à distorção causada pelo esticamento das coordenadas de textura U e V do objeto. Essa distorção pode aumentar o raio de atuação dos *blurs*, provocando um “embaçamento” exagerado em certas regiões assim como brilhos avermelhados em pontos perto das sombras.

De forma a amenizar esse problema d’Eon e Luebke (2007) propõem utilizar uma textura conhecida como *Stretch Texture* para armazenar um termo que regula o comprimento das gaussianas para cada localização no espaço da textura. Essa regulagem é feita através do uso das derivadas dos pontos tridimensionais nas coordenadas da tela (segundo d’Eon e Luebke (2007), a derivada pode ser considerada uma boa medida para verificar o quão esticada é uma parte da malha). Com isso, o tamanho das convoluções tende a ser igual ao redor de toda a malha e diferente no espaço da textura. A equação (27) mostra como calcular esse termo, o qual é utilizado na equação (25) para ajustar o raio de atuação das gaussianas ao redor de um pixel durante o processo de *blur*.

$$stretch_x = \frac{c}{ddx(p)} \quad (27)$$

onde c é uma constante de forma que os valores sejam escalados para a faixa $[0,1]$ (espaço da textura) e ddx é a derivada do ponto p na direção x . Para a direção y o cálculo é análogo.

A correção do esticamento realizado por essa textura auxiliar deve ser feita tanto para a coordenada U como para a V , uma vez que a convolução das gaussianas é realizada nas direções horizontal e vertical. A Figura 17 mostra um exemplo de *Stretch Texture* gerado enquanto que a Figura 18 mostra a correção obtida com o uso dessa textura.

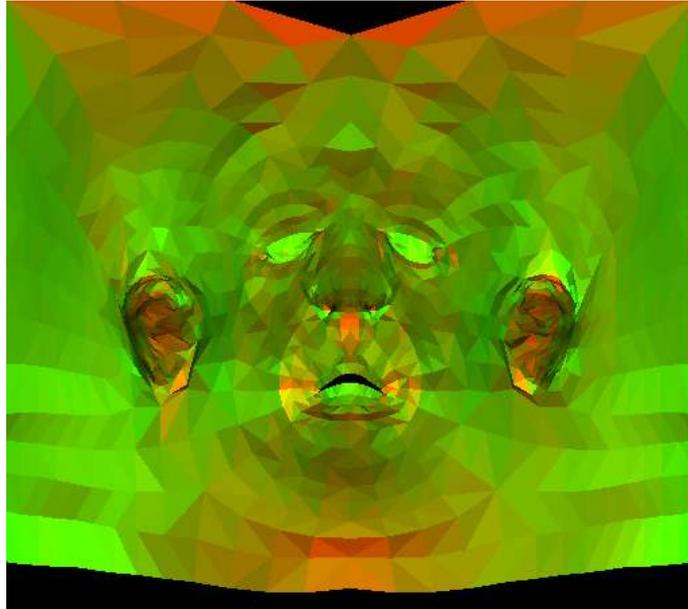
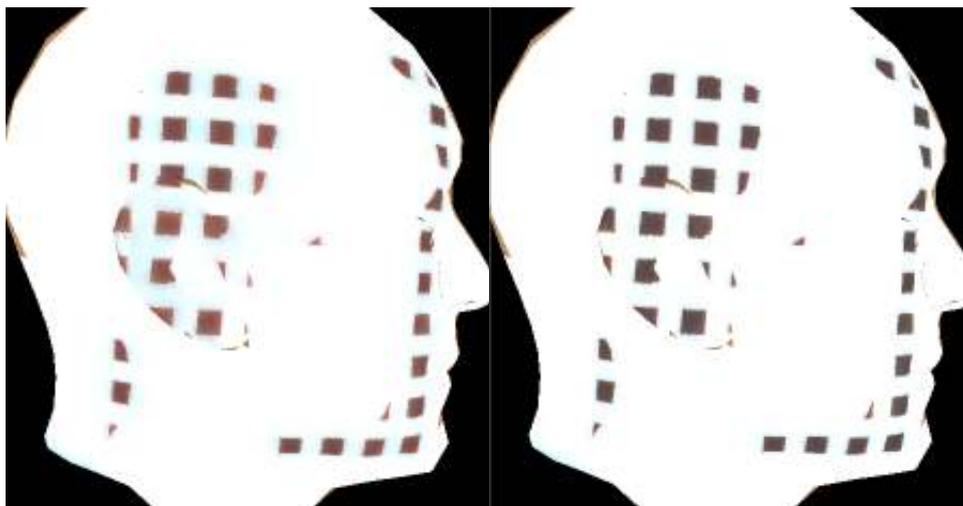


Figura 17 – Exemplo de *Stretch Texture*. Essa textura é uma imagem RGB onde na primeira coordenada é armazenada a correção na direção x e na segunda, a correção na direção y, conforme a equação (27).



(a)

(b)

Figura 18 – Aplicando certo padrão ao modelo utilizado, pode-se perceber o problema ocasionado devido a não correção da distorção causada pelo *unwrap* da malha. A imagem à esquerda (a) não possui correção alguma durante o processo de *blur* da textura de irradiância. Pode-se perceber que a orelha foi largamente afetada e “manchada”. Já na imagem à direita (b), a distorção foi ajustada de acordo com a região da face, amenizando assim borrões indevidos.

Conforme visto na Tabela 2, dependendo do comprimento da gaussiana, a área afetada pela convolução é maior. Quando a convolução é grande, a correção

feita pela textura auxiliar nem sempre é suficiente para acertar a distorção causada. Dessa forma, para complementar a correção do erro, d'Eon e Luebke (2007) sugerem que seja considerada a distorção da imagem convoluída. Para isso, ao invés de usar a mesma textura auxiliar (*Stretch Texture*) para criar cada textura de cada gaussiana, d'Eon e Luebke (2007) realizam uma convolução na *Stretch Texture* e utilizam-se essa nova textura auxiliar para gerar o resultado. A idéia, com isso, é utilizar um valor médio para diminuir o esticamento da imagem e corrigir a textura de irradiância convoluída que é gerada. Embora na teoria esse ajuste corrija de forma mais adequada problemas de distorções para grandes convoluções, d'Eon e Luebke (2007) informam que a diferença é praticamente imperceptível, podendo essa parte ser omitida do algoritmo, usando a cada passo apenas a *Stretch Texture* original.

4.2.5. Simulação de Não-Uniformidades da Pele

Nas seções 4.1.2 e 4.1.5 é explicado como é gerada a textura de irradiância e como deve ser incorporada a cor da pele após os *blurs* dessa textura para gerar o resultado final. Essa forma de junção de cor após o processo de *scattering* da luz não é de todo fisicamente correto.

A pele possui variações para baixas e altas frequências de cor, o que fisicamente corresponde a pequenas não-uniformidades nas diferentes camadas da pele que fazem com que o espalhamento e absorção da luz variem internamente. Na explicação dos perfis de difusão da seção 4.2.1, são considerados materiais homogêneos e uniformes, o que de fato a pele como um todo não é. Segundo d'Eon e Luebke (2007) a modelagem desse problema se torna impraticável em tempo real, sendo assim, eles propõem utilizar texturas para simular essa não-uniformidade.

Resumidamente, se a adição de cor for realizada antes do *blur* da textura de irradiância, ao final do processo muitos dos detalhes em alta frequência são perdidos; já se a adição for feita após as convoluções, não haverá espalhamento da cores diminuindo o aspecto de suavização existente no *scattering* da luz. d'Eon e Luebke (2007) propõem a utilização de uma escala do mapa de cor a ser aplicada antes do *blur* e outra escala após as convoluções. Essa simples idéia provoca tanto

o espalhamento da cor, suavizando um pouco a pele, como a manutenção dos detalhes de alta frequência existentes no mapa de cor original, simulando, assim um pouco da não-uniformidade da pele. A equação (10) pode então ser reescrita pela equação (28).

$$final_difuse = fac_shadow * difuse * C^{mix} \quad (28)$$

onde mix é a escala de cor, entre $[0,1]$, utilizada para gerar o mapa de irradiância.

A cor difusa final do algoritmo, gerada no último passo (seção 4.2.8), também deve considerar o fator mix (no caso $1-mix$ para não aumentar nem diminuir o fator de cor) conforme mostrado na equação (34). A Figura 19 mostra exemplos de resultados gerados utilizando a coloração antes e depois das convoluções. Já a Figura 20 mostra o resultado com um meio termo das opções. As Figuras 15 e 16 exibidas na seção 4.2.3 foram geradas utilizando essa técnica.



(a)

(b)

Figura 19 – A figura à esquerda (a) mostra um exemplo de coloração antes do *blur* do mapa de irradiância, enquanto que a figura à direita mostra um exemplo de coloração feita após as convoluções. A figura à esquerda apresenta uma pele com cor mais suavizada, porém perde detalhes da camada mais superficial da pele, como a barba, exibidos na figura à direita.



Figura 20 – Exemplo de resultado obtido utilizando um mix entre coloração antes e depois do *blur* da textura de irradiância.

4.2.6. Conservação de Energia

Até agora, as reflexões especular e difusa estão sendo tratadas separadamente, independente dos ângulos de entrada e saída da luz. Essa abordagem ignora uma sobreposição das duas no ponto de vista da energia envolvida. Uma mistura constante entre as duas reflexões, como é feito pela equação (19) ou pela forma mais genérica demonstrada na equação (29), não conserva energia, pois não considera a dependência da reflexão especular com o ângulo entre os vetores N e L .

$$k_s \text{specular} + k_d \text{diffuse} \quad (29)$$

onde k_s é o coeficiente especular do material e k_d o coeficiente difuso do mesmo.

A energia disponível para o espalhamento sob a superfície (parte difusa) é exatamente toda energia não refletida diretamente pela função de reflexão especular. Um simples cálculo de reflexão especular disponibiliza apenas uma porção do total de energia de entrada refletida em apenas uma direção de saída (uma direção L e uma direção V). Para obter o total de energia utilizada pela reflexão especular, todas as direções V ao redor do ponto analisado devem ser consideradas. Portanto, para obter a energia resultante para a parte difusa (ρ_{dt}) d'Eon e Luebke (2007) subtraem a energia total possível no ponto pelo resultado

da integral da reflexão especular para todas as direções do hemisfério conforme descrito na equação (30):

$$\rho_{dt}(x, L) = 1 - \int_{2\pi} f_r(x, \omega_0, L) (\omega_0 \cdot N) d\omega_0 \quad (30)$$

onde f_r é a função especular (no caso, a analisada na seção 4.1.4) e ω_0 os vetores no hemisfério sobre N .

A equação (30) pode ser transformada para coordenadas esféricas, conforme a equação (31) (a importância dessa transformação é descrita na seção 5.3 a qual contém detalhes de implementação).

$$\rho_{dt}(x, L) = 1 - \int_0^{2\pi} \int_0^{\frac{\pi}{2}} f_r(x, \omega_0, L) (\omega_0 \cdot N) \sin(\theta) d\theta d\phi \quad (31)$$

Reescrevendo a equação (28) de forma que a parte difusa contenha o termo de conservação de energia tem-se assim a equação (32).

$$final_difuse = \rho_{dt}(x, L) * fac_shadow * difuse * C^{mix} \quad (32)$$

Conforme visto anteriormente, uma parte da luz ao espalhar sob a superfície acaba saindo novamente por volta da região de entrada. Essa luz de saída sofre os mesmos efeitos de direção na camada externa da pele. Dependendo do ângulo em que a superfície é vista, uma quantidade diferente de luz difusa escapa da superfície da pele. Por esse motivo, d'Eon e Luebke (2007) realizam novamente a atenuação descrita acima após as convoluções serem efetuadas (conforme equação (34) na seção 4.2.8), tendo como diferença a utilização do vetor V ao invés do L para resolver a equação (30) (nesse caso a luz vem de todas as direções ao ponto de saída e sai na direção V).

4.2.7.

Mapa de Sombra Translúcido Modificado e o Espalhamento Global da Luz

O cálculo de difusão no espaço da textura captura o alto espalhamento local da luz, porém não obtém de forma completa o espalhamento global, nem a transmissão da luz, entre regiões muito finas que estão próximas no espaço tridimensional, mas distantes no espaço da textura. Esse espalhamento global é, por exemplo, responsável pela aparência avermelhada da orelha quando a observamos sem estar diretamente iluminada.

De forma a corrigir o problema supracitado, d'Eon e Luebke (2007) propõem modificar o mapa de sombra translúcida (*TSM*) (Dachsbacher e Stamminger 2003) de forma a permitir uma estimativa aproximada do espalhamento global com o auxílio das texturas de irradiância convoluídas.

Originalmente um *TSM* é uma textura que armazena para todos os pontos iluminados a distância z desse ponto à luz, sua normal N e sua cor difusa. No *TSM* modificado (d'Eon e Luebke 2007), são armazenadas a distância z e as coordenadas de textura U e V desses pontos. A idéia dessa abordagem é permitir que para cada ponto C em sombra possa ser calculada a distância até o ponto diretamente iluminado A (na direção do raio de luz) assim como o mapeamento desse ponto A nas texturas de irradiância convoluídas. O mapa de sombra translúcido é criado de forma semelhante ao *Shadow Map* explicado na seção 4.1.1, tendo como diferença apenas o armazenamento das coordenadas de textura U e V do ponto iluminado nas coordenadas G e B da textura RGB (na coordenada R é armazenada a distância).

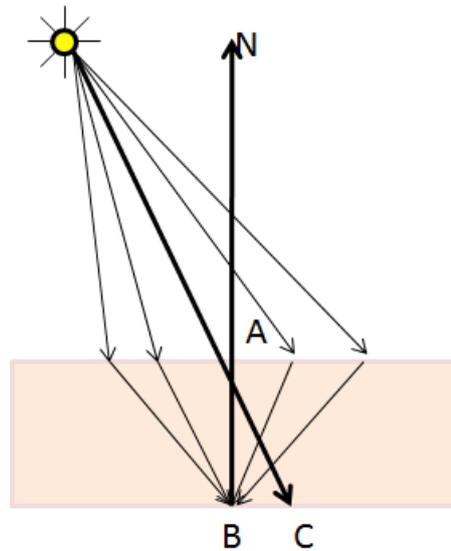


Figura 21 – Esquema do modelo matemático do espalhamento da luz pela pele até o ponto C.

Observando a Figura 21, pode-se modelar matematicamente como o espalhamento global da luz atua na pele humana. Considere um ponto C em sombra para o qual o raio de luz é interceptado pelo ponto A da pele. O objetivo é estimar o quanto a luz é espalhada pelos pontos próximos a A que chegam e iluminam o ponto C . Conforme visto anteriormente, a luz que chega em C se dá pelo perfil de difusão $R(m)$, onde m é a distância de C para cada ponto ao redor de A . Esse cálculo pode ser facilitado e aproximado se for utilizado o ponto B ao invés do ponto C . Pode-se fazer essa aproximação sem muitos problemas, pois para ângulos pequenos entre o raio de luz L e a normal N de A (θ), C fica muito próximo a B e para ângulos grandes, a reflexão especular é dominante na visualização, fazendo com que o erro seja imperceptível. O espalhamento em B para cada ponto a uma distância r de A (na superfície de A) se dá pela equação (33).

$$R\left(\sqrt{r^2 + d^2}\right) \approx \sum_{i=1}^k w_i G\left(v_i, \sqrt{r^2 + d^2}\right) = \sum_{i=1}^k w_i e^{\frac{-d^2}{v_i}} G\left(v_i, r\right) \quad (33)$$

onde $d = m \cdot \cos(\theta)$ (distância de B à A).

Nesse ponto vem a importância do uso do *TSM* modificado: os resultados dos cálculos de cada uma das gaussianas da equação (33) são armazenados nas

texturas de irradiância. O *TSM* modificado armazena as coordenadas de textura U e V do ponto iluminado A . Com isso, pode-se fazer o mapeamento nas texturas de irradiância para encontrar os resultados das gaussianas e utilizá-los para resolver a equação (33). Além disso, o *TSM* modificado guarda a distância do ponto A para a luz, permitindo assim o cálculo da distância m (distância de C menos a distância de A).

De forma a corrigir pequenos artefatos gerados por variações de alta frequência na distância armazenada no *TSM* modificado, d'Eon e Luebke (2007) propõem armazenar essa distância (corrigida pelo termo $\cos(\theta)$) na coordenada A do mapa de irradiância (o mapa de irradiância é uma textura RGBA) para que possa sofrer as mesmas convoluções as quais o mapa é imposto. Assim, quando o cálculo da equação (33) é feito para a gaussiana i , é utilizada a distância “embaçada” armazenada na textura de irradiância convoluída $i-1$. Os cálculos da equação (33) são realizados no último passo do algoritmo, descrito na seção 4.2.8.

As Figuras 22 e 23 mostram respectivamente um exemplo de mapa de sombra translúcida modificado e de espalhamento global da luz para regiões finas.

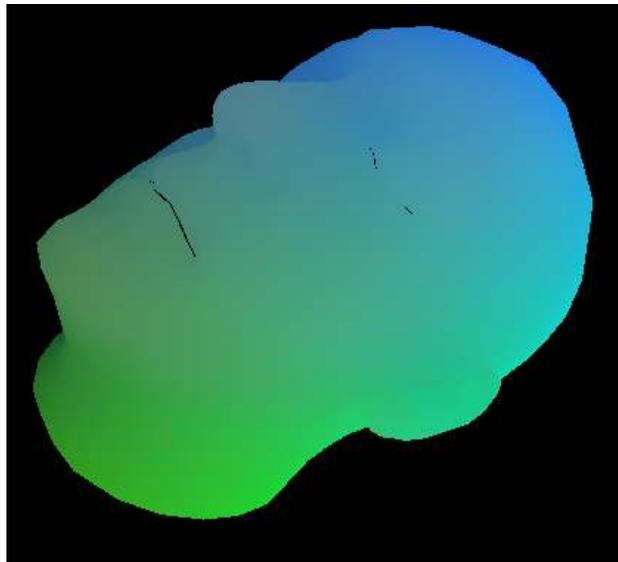


Figura 22 – Exemplo de *TSM* modificado. Assim como o *Shadow Map*, armazena a distância dos pontos iluminados até a luz na coordenada R . Nas coordenadas G e B , são armazenadas as coordenadas de textura U e V desses pontos.



Figura 23 – Exemplo de espalhamento global da luz. A parte de trás da orelha não sofre iluminação direta, porém devido ao *subsurface scattering* da pele, uma parte da luz chega e a ilumina.

4.2.8. Renderização Final da Malha

Assim como no algoritmo de Gosselin et al. (2004a), na última parte do algoritmo de d'Eon e Luebke (2007) é feita a junção das reflexões especular e difusa explicadas anteriormente para gerar a imagem final da pele humana. Nessa etapa, volta-se a trabalhar no espaço tridimensional, onde as texturas de irradiância convoluídas são combinadas linearmente e aplicadas à malha e o cálculo de reflexão especular é realizado de forma a obter o resultado final.

A combinação linear das texturas de irradiância convoluídas para gerar o termo difuso da pele se dá pela equação (34).

$$\begin{aligned}
 final_difuse_color = & C^{(1-mix)} \cdot \rho_{dt}(x, V) \cdot \\
 & \left(\sum_{i=1}^6 w_i \cdot final_difuse_blurred(i) + \right. \\
 & \left. \sum_{i=2}^6 w_i \cdot e^{-\frac{final_difuse_blurred(i-1) \cdot a^2}{v_i}} \cdot final_difuse_blurred(tsm_coord, i) \right)
 \end{aligned} \quad (34)$$

onde $final_difuse_blurred(i)$ é a cor do pixel para cada uma das texturas de irradiância convoluídas i (para a coordenada do pixel atual), $final_difuse_blurred(tsm_coord, i)$ é a cor do pixel para cada uma das texturas de irradiância convoluídas i na coordenada armazenada no mapa de sombra translúcida modificado, $final_difuse_blurred(i) \cdot a$ é a distância “embaçada” armazenada no canal A da textura de irradiância i e w_i os pesos de cada uma das gaussianas (Tabela 2).

A equação (34) já apresenta os termos de não-uniformidades na pele (seção 4.2.5), conservação de energia (seção 4.2.6) e de espalhamento global da luz pelas camadas internas da pele (seção 4.2.7) apresentados nas seções anteriores.

Assim como no algoritmo de Gosselin et al. (2004a), d’Eon e Luebke (2007) também consideram para gerar o termo especular as sombras na pele de forma que a mesma não apresente em regiões não iluminadas brilhos incompatíveis. d’Eon e Luebke (2007) utilizam o fator de sombra apresentado na equação (10) em conjunto com um fator de atenuação dado pelo ângulo entre a luz e a normal do ponto para gerar o termo especular final apresentado na equação (35). O modelo especular utilizado na equação (35) é o de Kelemen/Szirmay-Kalos (2001) apresentado na seção 4.1.4.

$$final_specular_color = specular * L_c * fac_shadow * (N.L) \quad (35)$$

Para gerar cada pixel da imagem final da malha com todo o algoritmo de simulação da pele humana apresentado por d’Eon e Luebke (2007) é utilizada a mesma equação (19) apresentada na seção 4.1.5 do algoritmo de Gosselin et al. (2004a). A Figura 24 mostra um exemplo obtido como resultado do uso dessa técnica.

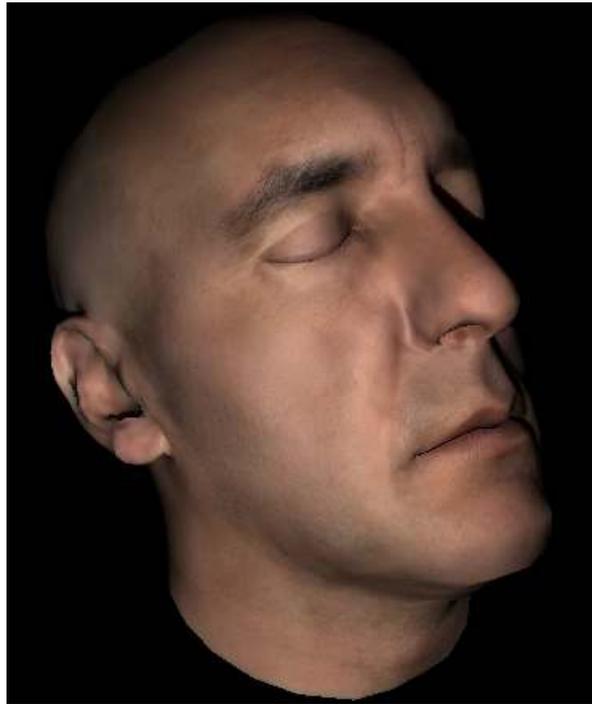


Figura 24 – Exemplo de resultado obtido utilizando a técnica de renderização de pele humana apresentada por d’Eon e Luebke (2007).

Assim como para a técnica de Gosselin et al. (2004a) apresentada em toda seção 4.1, nos capítulos 5 e 6 são exibidos mais resultados do uso da técnica de d’Eon e Luebke (2007), além de vários detalhes de implementação e um conjunto de vantagens e desvantagens observadas durante o desenvolvimento.